DOCUMENT RESUME

ED 084 877 EM 011 701

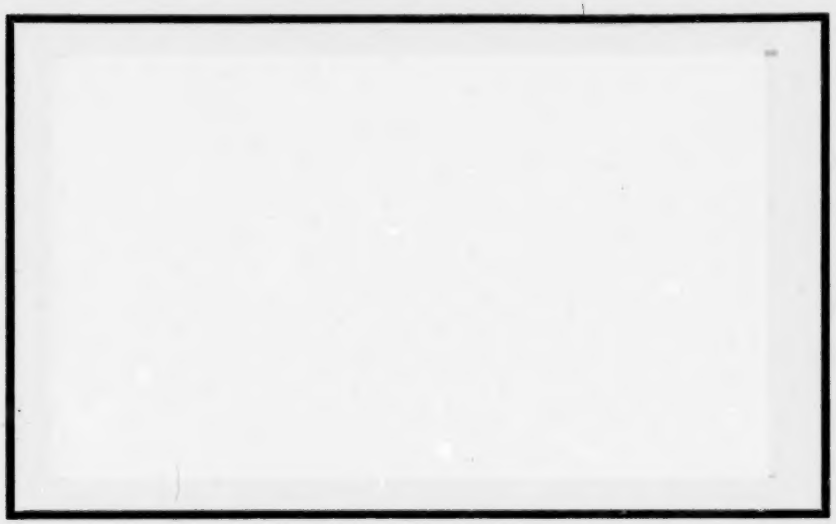
AUTHOR	Wheaton, Matthew; And Others
TITLE	Coursewriter II Functions for the Generation and Display of Plots and Other Graphics. Systems Memo Number Two.
INSTITUTION	Texas Univ., Austin. Computer-Assisted Instruction Lab.
SPONS AGENCY	Department of Defense, Washington, D.C. Advanced Research Projects Agency.; National Science Foundation, Washington, D.C.
REPORT NO	SM-2
PUB DATE	Oct 69
NOTE	50p.; See also EM 011 700 and EM 011 702
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Assisted Instruction; *Computer Graphics; Computer Oriented Programs; *Computer Programs; *Computer Science; Programing; Technical Reports; *Tutorial Programs
IDENTIFIERS	*Coursewriter II Functions; IBM 1510 Instructional System; Mathematical Functions; Plot System; Plotting

ABSTRACT

A series of six functions and a station command have been designed to enable Coursewriter II authors to use the plotting capability of the IBM instructional system in presenting course material. This capability has been augmented to include the ability to graph mathematical functions or other data on the 1510 terminal. Documentation of the Plot System is provided in this report, which is comprised of four sections. The Plot System User's Guide discusses usage and application rules for the Plot System, while the Programer's Guide contains the system logic, organization and installation procedures for use by systems programers. Section III presents a Tutorial for use with the Plot System demonstration program and the final part of the report consists of a Coursewriter II listing of the pilot system demonstration program. (Author/PB)

FILMED FROM BEST AVAILABLE COPY

ED 064877



THE UNIVERSITY OF TEXAS AT AUSTIN

Computer Assisted Instruction Laboratory

AUSTIN

EM 011701

ED 084877

U S DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRO
DUCED EXACTLY AS RECEIVED FROM
THE PERSON OR ORGANIZATION ORIGIN
ATING IT. POINTS OF VIEW OR OPINIONS
STATED DO NOT NECESSARILY REPRE
SENT OFFICIAL NATIONAL INSTITUTE OF
EDUCATION POSITION OR POLICY

COURSEWRITER II FUNCTIONS

FOR THE

GENERATION AND DISPLAY OF PLOTS AND

OTHER GRAPHICS

Systems Memo No. 2

October, 1969

Matthew Wheaton
Vaughn Groom
C. Victor Bunderson

Co-Sponsored By

The National Science Foundation
Grant GJ 509 X

and

Advanced Research Projects Agency
ARPA Order No. 1261
Contract N00014-67-A-0126-0006

Reproduction in whole or in part is permitted for any purpose of the United States Government.

This document has been approved for public release and sale; its distribution is unlimited.

Computer-Assisted Instruction Laboratory
The University of Texas at Austin
Austin, Texas 78712

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified

1. ORIGINATING ACTIVITY (Corporate author)
Advanced Research Projects Agency
Washington, D. C. 20301

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE
COURSEWRITER II FUNCTIONS FOR THE GENERATION AND DISPLAY OF PLOTS AND OTHER GRAPHICS

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Technical Report

5. AUTHOR(S) (First name, middle initial, last name)
M. Wheaton, V. Groom, & C. V. Bunderson

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1969	47	0

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
N00014-67-A-0126-0006	
b. PROJECT NO. NR 154-282/8-13-68 Code 458	N2-69
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U. S. Department of the Navy Office of Naval Research Washington, D. C.

13. ABSTRACT

A series of six functions and a station command have been designed to enable Coursewriter II authors to use the plotting capability of the IBM Instructional System in presenting course material. This capability has been augmented to include the ability to graph mathematical functions or other data on the 1510 terminal. This report provides documentation of the Plot System, including a User's Guide, System Programmer's Guide, Tutorial and Coursewriter II listing of the plot system demonstration. (U)

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified

Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Coursewriter II Functions						
Plot System						
IBM 1500 Instructional System						
User's Guide						
System Programmer's Guide						
Tutorial						
Plot Display						
Graphics						

PREFACE

This technical report provides documentation of the Plot System for use on the IBM 1500 Instructional System. It is comprised of four parts:

- Plot System User's Guide in which the usage and application rules for the Plot System are discussed from a user's standpoint.
- Systems Programmer's Guide which contains system logic, organization, and installation procedures for use by the systems programmer.
- Tutorial for use with the plot system demonstration program, "plot."
- Coursewriter II listing of the plot system demonstration program.

These Coursewriter II Functions were developed through funds provided by the Advanced Research Projects Agency of the Department of Defense and monitored by the Office of Naval Research under Contract No. N00014-67-A0126, and by National Science Foundation Grant GJ 509 X. The system analysis and programming was done by Mr. Matthew Wheaton of Booz-Allen Applied Research, Incorporated.

T A B L E O F C O N T E N T S

		Page
I.	PLOT SYSTEM USER'S GUIDE	I-1
1.	Limit Values	I-1
2.	Axis Placement	I-3
3.	Point Addition	I-4
4.	Plot Display	I-5
5.	Point Deletion	I-6
6.	Station Command PL	I-6
7.	Plot Transfer	I-7
II.	SYSTEM PROGRAMMER'S GUIDE	II-1
1.	Plot Directory and Point Storage	II-1
2.	System Installation	II-3
3.	Modification Notes	II-4

APPENDIX A: Tutorial for use with the plot demonstration program "plot."

APPENDIX B: Listing of demonstration program "plot."

I. PLOT SYSTEM USER'S GUIDE

The IBM 1500 Instructional System has the capability of rapidly presenting character oriented material on the 1510 display unit. This capability has now been augmented to include the ability to graph mathematical functions or other data on the 1510 terminal. A series of six functions and a station command have been designed to enable Coursewriter II authors to use this plotting capability in presenting course material.

The plot system uses an area of the display which is 30 columns by 30 rows. This area is in the upper right portion of the display. The leftmost ten columns and the bottom two rows are thus left for author use in displaying explanatory or descriptive material. Thirty rows by 30 columns i.e., 240 X 180 points are used for the plot area since there are 8 points per column and 6 points per row. Each point may be individually addressed to allow the author complete control over the display.

1. LIMIT VALUES

Before specifying any point to be turned on, the author must set the limit values in both the X and the Y directions. Function SET has been provided to do this in coordination with function FCALC. The

X limit values describe the values of the leftmost and rightmost points on the horizontal scale and the Y limit values describe the values of the bottommost and topmost points of the vertical scale.

All values are floating point. Hence, any numerical expression acceptable to function FCALC is valid. Two parameter values are provided for function FCALC to use in conjunction with the plot system. They are:

- fn fcalc/'5 where numeric evaluation is made of the contents of buffer 0. A result is returned in display code to buffer 0. In addition, the resulting value, in floating point format, is placed in counters 29 and 30. A request to raise a negative base to a non-zero power will be treated as if the parameter were 0.

- fn fcalc/'6 where numeric evaluation is made of the contents of buffer 0. A result is returned in display code to buffer 0. In addition, the resulting value, in floating point format, is placed in counters 29 and 30. A request to raise a negative base to a non-zero power will be treated as if the parameter were 1.

Function SET uses the contents of counters 29 and 30 to store limit values for the X or Y scales depending on the parameter of the call. Function SET action is as follows:

- fn set/'xs stores the contents of counters 29 and 30 as the leftmost coordinate for the display of the active station.

- fn set/'xe stores the contents of counters 29 and 30 as the rightmost X coordinate for the display of the active station.

- fn set/'ys stores the contents of counters 29 and 30 as the bottommost Y coordinate for the display of the active station.

- fn set/'ye stores the contents of counters 29 and 30 as the topmost Y coordinate for the display of the active station.

- If any other parameter is used, function SET will return error code 25.

The following set of instructions will describe the plot area such that all points with X coordinate greater than or equal to -100 and less than or equal to 100 and with Y coordinate greater than or equal to 20 and less than or equal to 50 will lie within the graph.

```
ld    -100/'b0          ld    20/'b0
fn    fcalc/'5          fn    fcalc/'5
fn    set/'xs           fn    set/'ys
ld    100/'b0           ld    50/'b0
fn    fcalc/'5          fn    fcalc/'5
fn    set/'xe           fn    set/'ye
```

2. AXIS PLACEMENT

After the limit values are set, the placement of the coordinate axes may be set. Function AXIS is prepared for this purpose. Its action is as follows:

- fn axis/'d sets a flag so that no coordinate axes are displayed at the active station when function PLOT is used.

- fn axis (null parameter, or any other than d) uses the contents of counters 27 and 28 as the floating point value of the X coordinate of the axes intersection, and the contents of counters 29 and 30 as the floating point value of the Y coordinate of the axes intersection for the active station. Whenever function PLOT is used, horizontal and vertical axes will be generated which intersect at the indicated point.

If the axis limits have not been set, or set improperly, such as both limits at the same value, function axis will not set any axis pointers but will store an error code in counter 23. These codes are the same as described in the documentation of function FCALC. The contents of counter 22 will be meaningless after an error.

If the limits had been set as described above, the following instructions would cause the axes to intersect at the center of the plot area.

```
ld    0/b0
fn    fcalc/5
ld    c29/c27
ld    c30/c28
```

```
ld    (20+50)/2/b0
fn    fcalc/5
fn    axis
```

3. POINT ADDITION

Once the limit values have been set, it is possible to turn on any point in the plot display area by describing its coordinate values. Function ADDPNT is prepared to store points for the active terminal. Function ADDPNT has no parameter. When it is called, the contents

of counters 27 and 28 are used as the floating point value of the X coordinate of the point and the contents of counters 29 and 30 are used as the floating point value of the Y coordinate of the point. As in function AXIS, an arithmetic error will result in no point stored for the active terminal, an error code stored in counter 23, and the contents of counter 22 destroyed.

If the limits had been set as described above, the following instructions will turn on the points at the four corners of the display.

```
ld    -100/'b0          ld    100/'b0
fn    fcalc/'5          fn    fcalc/'5
ld    c29/'c27          ld    c29/'c27
ld    c30/'c28          ld    c30/'c28
ld    20/'b0            ld    20/'b0
fn    fcalc/'5          fn    fcalc/'5
fn    addpnt            fn    addpnt
ld    50/'b0            ld    50/'b0
fn    fcalc/'5          fn    fcalc/'5
fn    addpnt            fn    addpnt
```

If at any time the space for the active terminal is depleted, i.e., over 959 points are stored, function ADDPNT returns message code P1. If no disk space is available for any terminal, the function returns message code P2.

4. PLOT DISPLAY

When one or more points have been set for the active terminal by use of ADDPNT, the plot may be displayed. This is done by using

function PLOT. This function, which has no parameters, creates a display of the points which have been set by ADDPNT and the axes which have been set by AXIS. No error messages are generated by PLOT. However, if there are no points to be displayed, function PLOT returns with no action taken.

5. POINT DELETION

When an author begins to build a display, and when one is no longer needed for the active station, function DELETE should be used. This function, which has no parameter, deletes all the stored points for the active terminal and restores disk space for use by other terminals or for re-use by the active terminal. No error messages are generated by function DELETE.

6. STATION COMMAND PL

Plots may be built up dynamically using Coursewriter II instructions as has been described above. In addition to this, an author may build a display at his terminal, name it, and recall it by name at 1 or more student terminals. The station command PL has been provided to assist the author in doing this. To use it, the author first builds a plot as desired at his terminal using the above described functions. The he gets system attention and types pl a,xxxxxx, where xxxxxx is a 1 to 6 character name. The station command will store the active terminal's plot under

the given name, delete it from the active terminal, and return the message "command completed."

If a named plot is no longer needed, an author may get attention, and type pl d,xxxxxx where xxxxxx is the name of a stored plot. The station command will delete the plot, add its space to the free area, and return the message "command completed."

An author may view the names of the stored plots by getting attention, and typing pl l. The command will return a list of all plots stored by name.

7. PLOT TRANSFER

A named plot may be retrieved at any station by use of the function TRNSFR. This function, which has no parameter, uses the first 6 characters of buffer 0 as the name to search for in the table of stored plots. When the plot is found, a copy is made in the storage area for the active terminal, and any plot there is released as free space for use by any terminal. The following instructions would bring a plot named "abcdef" into the active terminal's storage area.

```
ld    abcdef/b0                    fn    trnsfr
```

If the above named plot were not in the table of saved plots, the function would return error code P3. If there is no room available in which to store the plot, the function returns error code P4.

II. SYSTEM PROGRAMMER'S GUIDE

1. PLOT DIRECTORY AND POINT STORAGE

Four disk sectors are required to keep a directory of plot activity. They may be on any logical pack. The pack number must be indicated as LOGPK when assembling the six functions and the station command. In addition, the first sector of the directory must be indicated as DRTRY in the assemblies.

(1) Sector 1 Layout

- Words/0 - /F contain, by terminal, the number of data points stored.
- Words /20 - /3F contain, by terminal, the sector number of the first sector of stored data points.
- Words /40 - /5F contain, by terminal, the sector number of the second sector of stored data points.
- Words /60 - /7F contain, by terminal, the sector number of the third sector of stored data points.
- Word /80 is always zero.
- Words /81 - /DF contain a list of sectors which are available to store data points. This is known as the free table.
- Word /EO contains the address within the sector of the last entry in the free table.

- Words /100 - /11F contain a value which is the number of points from the left of the plot area where the Y axis will be located.

- Words /120 - /13F contain a value which is the number of points from the top of the plot area where the X axis will be located. If the value is /FF, no axis will be displayed.

(2) Sector 2 Layout

- Words /0 - /3F contain, by terminal, two word entries in floating point format indicating the leftmost limit of the plot.

- Words /40 - /7F contain, by terminal, two word entries in floating point format indicating the rightmost limit of the plot.

- Words /80 - /BF contain, by terminal, two word entries in floating point format indicating the bottommost limit of the plot.

- Words /C0 - /FF contain, by terminal, two word entries in floating point format indicating the topmost limit of the plot.

(3) Sectors 3 and 4 Layout

Sectors 3 and 4 of the directory contain the table of named plots. They are broken into eighty groups of eight words each. Each of these eight word groups is either entirely zero or:

- Words /0 - /2 contain 6 EBCDIC characters comprising the name, left adjusted blank filled.

- Words /3 - /5 contain the addresses of the sectors which hold the plot data.

- Word /6 contains the count of data points stored for the plot.

- Word /7 contains the coordinate points for the axes intersection. The left byte contains the number of points from the top of the plot area where the X axis will be displayed. If /FF is found there, no axes will be displayed. The right byte contains the number of points from the left of the plot area where the Y axis will be displayed.

(4) Point Storage

All other sectors, which must first be in the free table, contain data points for terminals or for named plots in the save table. Each word is in the following format:

- The left byte contains the number of points from the top of the display area where the point will be displayed.

- The right byte contains the number of points from the left of the display area where the point will be displayed.

- The last word of a set of data points contains /FFFF as a flag.

2. SYSTEM INSTALLATION

An area of from 16 to 96 sectors (1 to 6 blocks) should be reserved on a single disk pack. This is best done by registering courses and then changing the first character of the course name in the directory to a blank. This will leave the area available for data points and prevent inadvertant deletion of the area.

Four contiguous sectors of this area should then be used as the directory. To prepare them, the system programmer should insure that all four sectors are filled with zeros. Then, starting with word /81 of the first sector, he should put in the addresses of all sectors which are available for point storage. This should include all sectors which have been reserved except the four to be used as the directory. Lastly, word /E0 should be set to contain the address of the word with the last sector entry in the free table.

After setting up the directory, the six functions and the station command should be assembled. The EQUs referenced by the names LOGPK, DRTRY, and FLSEK should be set respectively to the value of the directory logical pack, the directory first sector address and the sector address where the absolute overlay of FCALC is loaded.

The functions should then be loaded and the station command put on the system pack according to standard procedures. The station command monitor should be patched to include the letters "PL" and the sector address of the station command.

3. MODIFICATION NOTES

Functions AXIS and ADDPNT interface with the overlay portion of function FCALC. It is imperative that appropriate modification be made to the EQUs in these functions if FCALC is changed.

Version two of FCALC must be used so that parameters 5 and 6 return a value to counters 29 and 30.

If only object decks are available for installation, install them as directed above and then (1) store the logical pack number in word /2B of Plot, word /71 of TRNSFR, word /28 of DELETE, word /61 of AXIS, word /2 of SET, and words /3 and /D3 of ADDPNT; (2) store the address of the first sector of the directory in word /4B of PLOT, word /70 of TRNSFR, word /3 of DELETE, word /4E of AXIS, word /8 of SET and word /5 of ADDPNT; (3) store the address of the third sector of the directory in word /73 of TRNSFR; (4) store the address of the FCALC overlay in word /63 of AXIS and word /E3 of ADDPNT; and (5) patch the station command by setting word /7230 to the logical pack number, word /7227 to the first sector of the directory and word /722F to the third sector of the directory.

A P P E N D I X A

TUTORIAL FOR USE WITH THE PLOT DEMONSTRATION PROGRAM "PLOT"

1.0 Coursewriter II Plot Functions

The purpose of this document is to familiarize authors with a set of plot functions available for use with Coursewriter II. To understand the use of the six graph functions and one station command which are used in connection with the definition and utilization of graphical information, the following should be kept in mind. First, there are three aspects of a graph which can be manipulated independently of one another: (1) the scale of the x and y dimensions, (2) the origin of the axis with respect to the scale, and (3) the coordinates of a point with respect to the scale.

In the second place, information defining a graph can be thought of as residing in two places. One of these is in a buffer assigned to a given terminal; the other is in a buffer referred to as the graph library. Entries in the graph library can only be changed by the station command. Information can be read from the library or manipulated in the terminal buffer with the six graph functions.

The "plot" demonstration program provides a convenient way to work with graphical information as an author at the terminal. The program incorporates some of the logic which an author will have to explicitly provide in courses developed by him. It is suggested that this document be used while signed on to the demonstration program "plot". One possible sequence through this program is given in Table 1.1 with corresponding references to relevant pages in the documentation. To follow this sequence, perform the action(s) listed in the first column and then refer to the corresponding section of the document given in the second column.

Table 1.1 Suggested Sequence through Plot Demonstration Program

1. TO SIGN ON:	
on plot/ ⟨author number⟩	(GO TO NEXT ACTION)
2. TO EXECUTE PROGRAM IF YOU HAVE NOT SIGNED ON THIS COURSE BEFORE:	
ex plota °/ e	section 2.0
OTHERWISE:	
ex plot °/ e	
3. TO EXECUTE FN SET, ENTER:	
s	section 3.0
4. ENTER AS REQUESTED:	
0 for x axis start value	
100 for x axis end value	
0 for y axis start value	
100 for y axis end value	(GO TO NEXT ACTION)
5. TO EXECUTE FN AXIS, ENTER:	
a	section 5.0
6. ENTER AS REQUIRED:	
no for "Do you wish to delete axis?"	
0 for x coordinate of y axis	
0 for y coordinate of x axis	section 6.0
7. TO EXECUTE FN ADDPNT, ENTER:	
m	section 7.0

(Note: an alternate coded slash is indicated by this symbol: °/)

8. ENTER AS REQUIRED:	
0 for x coordinate value	
0 for y coordinate value	section 8.0
9. TO EXECUTE FN AXIS, ENTER:	
a	section 9.0
10. TO EXECUTE FN AXIS, ENTER:	
a	
no for "Do you wish to delete axis?"	
200 for x coordinate of y axis	
200 for y coordinate of x axis	section 10.0
11. TO EXECUTE FN SET, ENTER:	
s	
0 for x axis start value	
400 for x axis end value	
0 for y axis start value	
400 for y axis end value	section 11.0
12. TO EXECUTE FN ADDPNT, ENTER:	
m	section 12.0
13. TO EXECUTE FN DELETE, ENTER:	
d	section 13.0
14. TO EXECUTE FN TRANSFER, ENTER:	
t	
funfun for "Enter plot name"	section 14.0
15. TO EXECUTE FN TRANSFER, ENTER:	
t	
funfun for "Enter plot name"	section 15.0

16. TO EXECUTE pl 1, ENTER:	
[ATTENTION]	
qu	
[ATTENTION]	
pl 1	section 15.0
17. TO EXECUTE pl a °/ graph name AND LIST GRAPH LIBRARY, ENTER:	
[ATTENTION]	
qu	
[ATTENTION]	
pl a °/ my own	
[ATTENTION]	
pl 1	section 17.0
18. TO EXECUTE pl d °/ graph name AND LIST GRAPH LIBRARY, ENTER:	
[ATTENTION]	
pl d °/ my own	
[ATTENTION]	
pl 1	section 18.0

2.0 The "Plot" Demonstration Program

This program has two basic purposes:

(1) provide a demonstration of the six plot functions

(2) provide an environment for an author to define graphs for the graph library which may be referred to in CAI courses.

The capabilities of the plot functions will be illustrated first. Compare Figure 2.1 with the screen display. Information for the plot area can be manipulated using:

(1) fn set (set scale for x and y dimensions)

(2) fn axis (define origin for axis)

(3) fn addpnt (add a point to the graph)

(4) fn plot (plot graph using information stored about scale, axis and points)

(5) fn delete (delete graph)

(6) fn trnsfr (transfer predefined graph from graph library to terminal buffer)

GO TO ACTION 3, TABLE 1.1

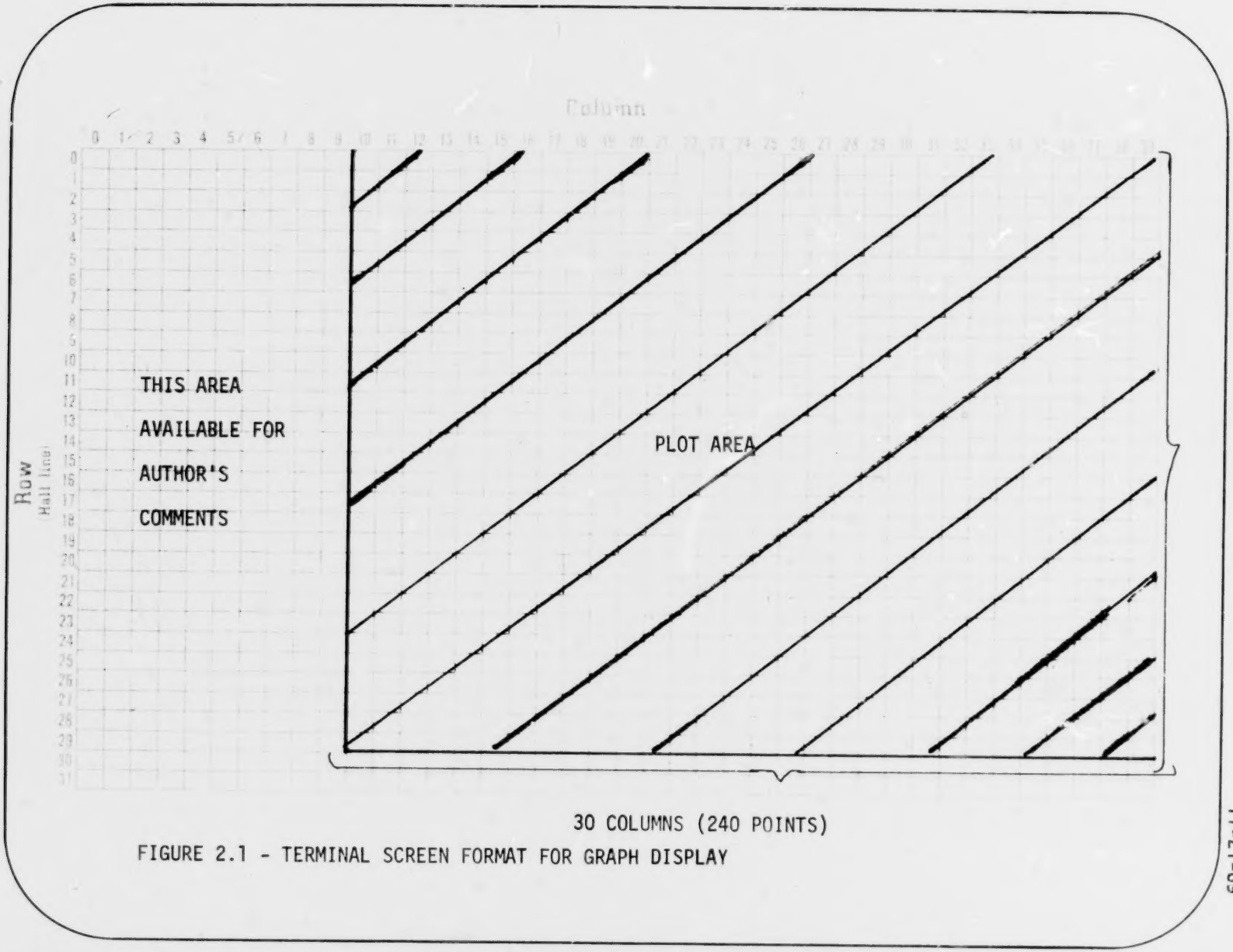


FIGURE 2.1 - TERMINAL SCREEN FORMAT FOR GRAPH DISPLAY

3.0 The "Set Scale" Function

This function provides a definition of the scale assigned to the x and y dimensions of the graph displayed in the plot area. The format of this function is:

fn set °/ { xs / xe / ys / ye } ≠

where

xs = set x dimension start of scale

xe = set x dimension end of scale

ys = set y dimension start of scale

ye = set y dimension end of scale

Prior to the execution of the set function, counters 29 and 30 must be loaded with the desired scale value in floating point form. This can be done by loading the numerical value (or numerical expression) into b0 and then executing:

fn fcalc °/ 5 ≠

The floating point version of the value in b0 will be returned in c29 and c30. An example of the use of fn set in the demo program is given in Figure 3.1. Note that the demo program requires you to enter only the actual numerical value or numerical expression for the x and y dimensions.

GO TO ACTION 4, TABLE 1.1

```
SETAX*E
      1 PR   *E
      2 DE   0+/32*E
      3 DT   ,0+/2,+/40,0+/(G)IVE X AXIS START VALUE,*E
      4 EP   ,0+/2,+/40,0+/+/99+/*E
      5 AA   (8+/)AA*E
      6 LD   0+/C23*E
      7 FN   FCALC+/5*E
      8 BR   SETAX+/C23+/G+/0*E
      9 FN   SET+/XS*E
SETXE*E
      1 DT   ,0+/2,+/40,0+/(G)IVE X AXIS END VALUE*E
      2 LD   0+/C23*E
      3 EP   ,0+/2,+/40,0+/+/99+/*E
      4 AA   (8+/)AA*E
      5 FN   FCALC+/5*E
      6 BR   SETXE+/C23+/G+/0*E
      7 FN   SET+/XE*E
SETYS*E
      1 DT   ,0+/2,+/40,0+/(G)IVE (Y )AXIS START VALUE *E
      2 LD   0+/C23*E
      3 EP   ,0+/2,+/40,0+/+/99+/*E
      4 AA   (8+/)AA*E
      5 FN   FCALC+/5*E
      6 BR   SETYS+/C23+/G+/0*E
      7 FN   SET+/YS*E
SETYE*E
      1 DT   ,0+/2,+/40,0+/(G)IVE (Y )AXIS END VALUE*E
      2 LD   0+/C23*E
      3 EP   ,0+/2,+/40,0+/+/99+/*E
      4 AA   (8+/)AA*E
      5 FN   FCALC+/5*E
      6 BR   SETYE+/C23+/G+/0*E
      7 FN   SET+/YE*E
      8 BR   PLOT*E
```

FIGURE 3.1 - Use of fn set and associated logic in the "plot" demonstration program

5.0 The "Define Axis" Function

The effect of this function is to define the origin of the axis or to delete the axis. The origin may be any value within the coordinate scales established by the set function. The function format is:

fn axis °/ d

where:

d= delete current axis (not used when defining axis)

Prior to the execution of the function, counters 27 and 28 must be loaded with the x coordinate of the y axis in floating point form. Likewise, counters 29 and 30 must be loaded with the y coordinate of the x axis in floating point form. The author would use:

fn fcalc °/ 5 ≠

in this connection as described in section 3.0.

An example of the use of fn axis in the demo program is given in Figure 5.1.

GO TO ACTION 6, TABLE 1.1

```
AXIS*E
      1 PR   *E
      2 DE   0+/32*E
      3 DT   ,0+/2,+/40,0+/(D)O YOU WISH TO DELETE AXES(/*E
      4 EP   ,0+/2,+/40,0+/+/3+/*E
      5 CA   YES+/CC*E
      6 FN   AXIS+/D*E
      7 BR   PLOT*E
      8 AA   (8+/)AA*E
AXISX*E
      1 DT   ,0+/2,+/40,0+/(E)NTER (X )COORDINATE OF (Y )AXIS.*E
      2 LD   0+/C23*E
      3 EP   ,0+/2,+/40,0+/+/99+/*E
      4 AA   (8+/)AA*E
      5 FN   FCALC+/5*E
      6 BR   AXISX+/C23+/G+/0*E
      7 LD   C30+/C28*E
      8 LD   C29+/C27*E
AXISY*E
      1 DT   ,0+/2,+/40,0+/(E)NTER (Y )COORDINATE OF (X )AXIS. *E
      2 LD   0+/C23*E
      3 EP   ,0+/2,+/40,0+/+/99+/*E
      4 AA   (8+/)AA*E
      5 FN   FCALC+/5*E
      6 BR   AXISY+/C23+/G+/0*E
      7 FN   AXIS*E
      8 BR   PLOT*E
```

FIGURE 5.1 - Use of fn axis and associated logic in the "plot" demonstration program

6.0

Every time the screen shows the current display, the demo program calls fn plot. Note, however, that the plot area is blank even though we have established a scale and defined an axis. This illustrates the requirement that AT LEAST ONE POINT MUST BE DEFINED BEFORE ANY PORTION OF A GRAPH IS DISPLAYED. If it is desired to display an axis only, a point could be defined on the axis and thus it would be overlayed by the axis line.

GO TO ACTION 7, TABLE 1.1

7.0 The "Add Point" Function

This function stores information defining a point for use by the plot function at a later time. The format is:

fn addpnt ≠

Prior to the execution of this function counters 27 and 28 must be loaded with the x coordinate of the point in floating point form. Likewise, counters 29 and 30 must be loaded with the y coordinate of the point in floating point form. The author would use:

fn fcalc °/ 5 ≠

in this connection as described in section 3.0.

An example of the use of fn addpnt in the demo program is given in Figure 7.1

GO TO ACTION 8, TABLE 1.1

```
POINT*E
     1 PR   *E
     2 DE   0+/32*E
     3 DT   ,0+/2,+/40,0+/(E)NTER (X )COORDINATE VALUE*E
     4 EP   ,0+/2,+/40,0+/+/99+/*E
     5 LD   0+/C22*E
     6 LD   0+/C23*E
     7 FN   FCALC+/5*E
     8 BR   POINT+/C23+/G+/0*E
     9 LD   C30+/C28*E
    10 LD   C29+/C27*E
POINT2*E
     1 PR   *E
     2 LD   0+/C22*E
     3 LD   0+/C23*E
     4 DT   ,0+/2,+/40,0+/(G)IVE (Y )COORDINATE VALUE*E
     5 EP   ,0+/2,+/40,0+/+/99+/*E
     6 FN   FCALC+/5*E
     7 BR   POINT2+/C23+/G+/0*E
     8 FN   ADDPNT*E
     9 BR   PLOT*E
```

FIGURE 7.1 - Use of fn addpnt and associated logic in the "plot" demonstration program

8.0

Recall that the demo program calls fn plot whenever it returns to this place in the program. Note that the axis is now displayed. (The point defined at (0,0) has been overlayed by the axes.)

The next sequence of actions will demonstrate that the definition of the axes is independent of the points plotted.

GO TO ACTION 9, TABLE 1.1

9.0

Recall that the plot scale is still 0-100 for both the x and y dimensions. Using the demo program, enter several new values for the origin of the axes and note the effect. (Remember to enter "a" for axis when the "what next" statement appears in the lower right hand corner of the screen.)

When you have satisfied your curiosity about this feature, return to the place in the program where the graph is plotted and then:

GO TO ACTION 10, TABLE 1.1

10.0

In this case the origin of the axis is defined outside the scale limits originally established. Note the appearance of the axis origin at (100,100). Definition of an origin outside the scale limits will cause the axis to appear at those limits.

A change in scale using fn set will be made next.

GO TO ACTION 11, TABLE 1.1

11.0

Note that the change of scale did not alter the axis position. To define the origin at (200,200) with respect to the new scale established by fn set requires redefinition of the origin.

Perhaps you would like to try other combinations of fn set and fn axis. As you do, notice that numerical expressions such as 10 + 10, 5 x 2, etc., can be used. The reason for this is that fn fcalc is used to get the data into the proper format. When you finish, return the scale values to:

0 for x and y axis start value

100 for x and y axis end value

and return the axis origin to (0,0).

GO TO ACTION 12, TABLE 1.1

12.0

Enter both numerical values and numerical expressions for the x and y coordinates for several points. You may also wish to move the axis about, demonstrating the independence of the plotted points and the axis. Define a point outside the scale range to see what happens.

GO TO ACTION 13, TABLE 1.1

13.0

Note that the plot area on the screen is blank. This resulted from using the delete function where the format is:

fn delete ≠

The basic capability has been demonstrated for five of the six plot functions: fn set, fn axis, fn plot, fn addpnt, and fn delete. The next phase of the explanation describes the transfer function (fn transfer) and illustrates the use of the "plot" demo program to define an entry in the graph library.

The description at this point is functional rather than specific. Documentation will be available at a later date for those who want to know the details of how the system handles information about the graph.

Imagine there exists for each terminal a graph buffer as shown in Figure 13.1. As indicated in the Figure, fn plot uses the information in this buffer to provide a graph on the screen. The information was put into the buffer by three of the functions you have used so far. (The delete function, obviously, deletes the information in this buffer.)

But there is another possible source of information for the terminal graph buffer other than the three functions mentioned above. A predefined graph in the graph library (we will consider how it was defined later), and the transfer function are involved here.

As shown in Figure 13.2, a copy of a graph in the graph library may be transferred to the graph buffer at the terminal used by a given student. The graph in the terminal buffer can then be updated through the plot functions based on the way the student progresses through the course. The original graph in the graph library remains unchanged as do the copies of the graph which might be in other terminal buffers.

Thus the transfer function:

fn trnsfr °/ <graph name> ≠

is used to move the information from the graph library into the terminal graph buffer. For a demonstration of the use of the transfer function, assume that a graph named funfun exists in the graph library.

GO TO ACTION 14, TABLE 11

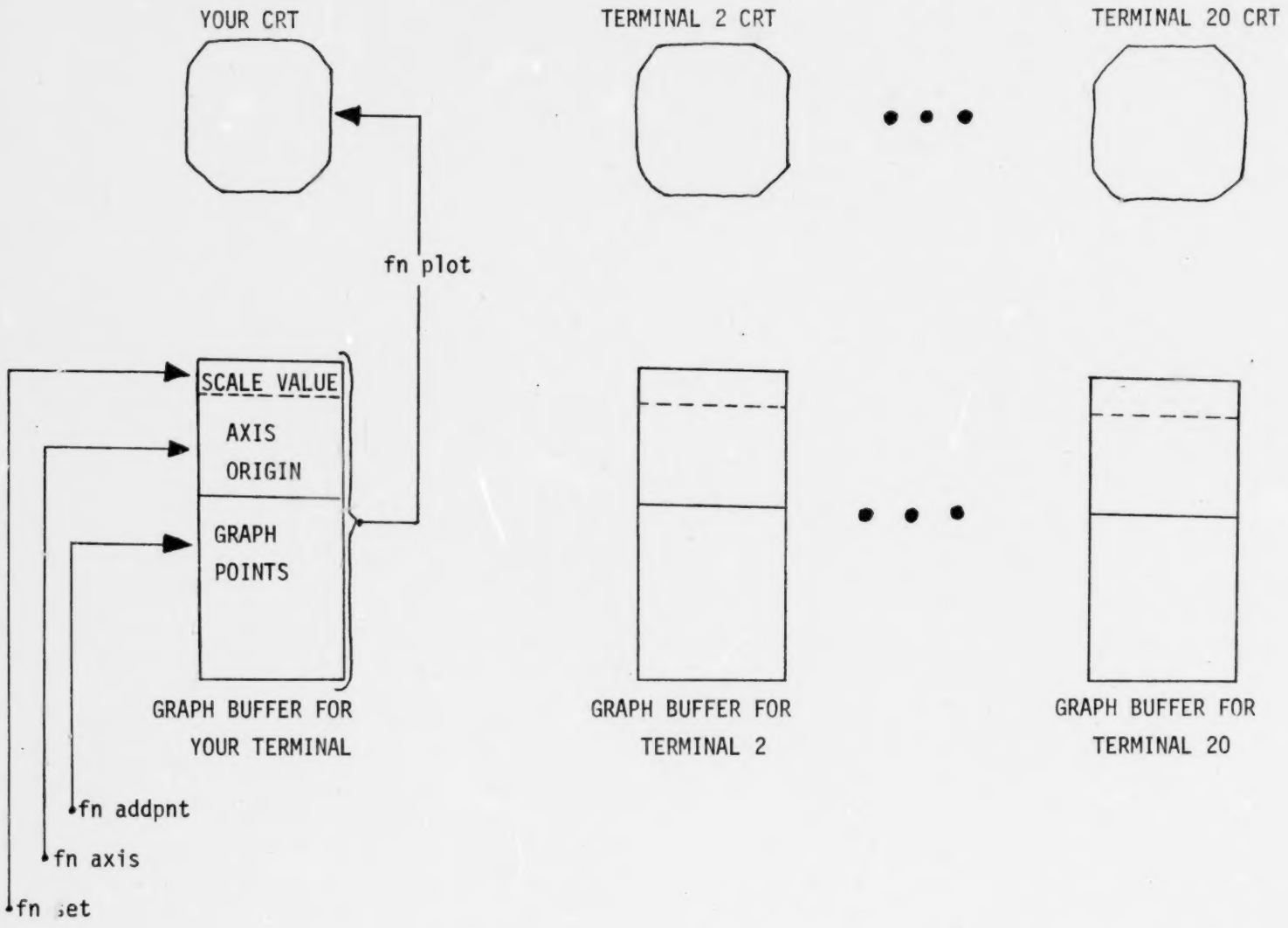


FIGURE 13.1 A graph buffer exists for each student terminal (functional representation)

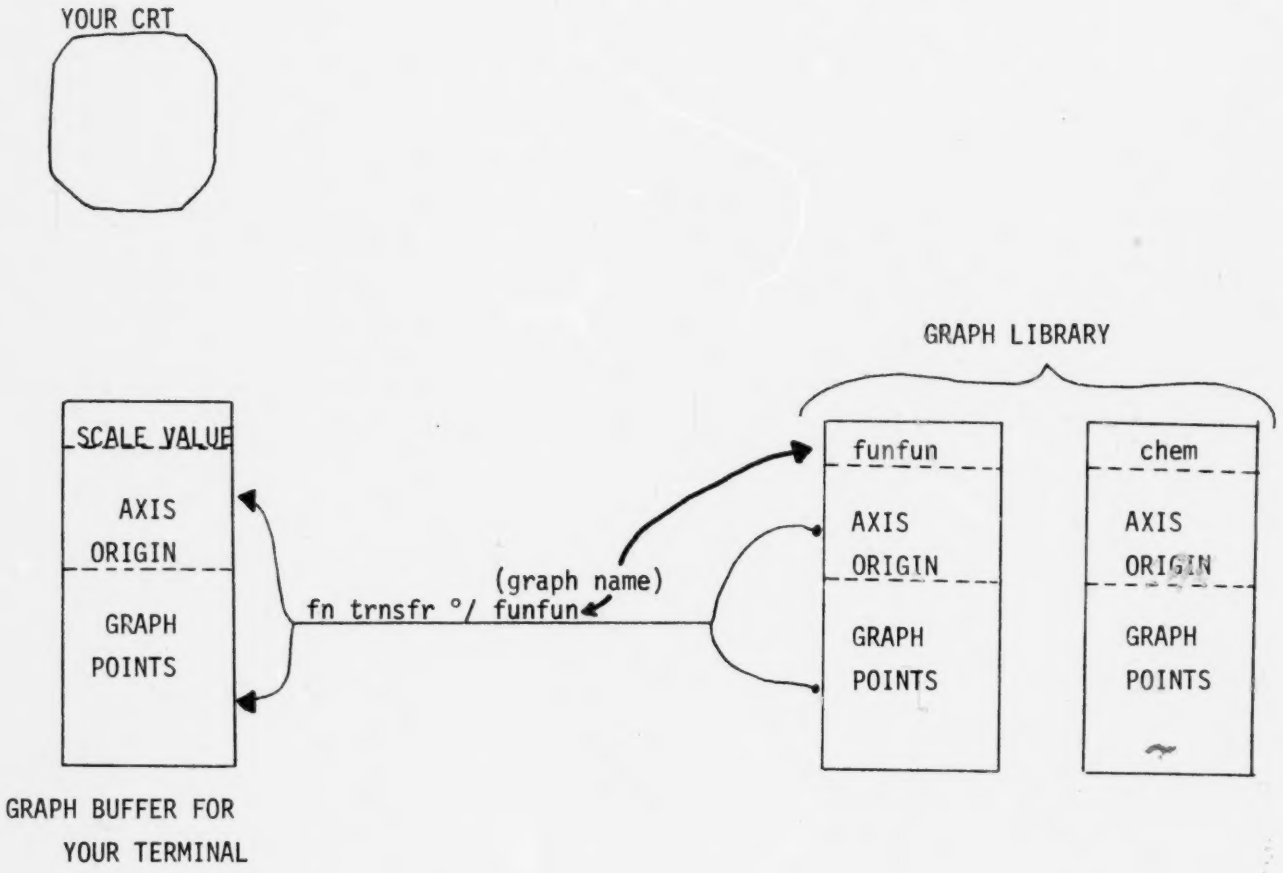


FIGURE 13.2 Transferring information from graph library to terminal buffer (functional representation)

14.0

Information about the graph funfun was transferred to your terminal's graph buffer. Enter additional points in the same way you did earlier and then·

GO TO ACTION 15, TABLE 1.1

15.0

Notice that the original graph, funfun, has overlayed the information in your terminal buffer.

Now we will discuss how to define a graph in the graph library (i.e., transfer information from the terminal buffer to the graph library). A station command, pl, is used in this situation. The format of this command is:

pl {a / d / l} °/ <graph name>

where:

a is the add option for entering <graph name> into the library

d is the delete option for deleting <graph name> from the library

l (as the only parameter) yields a listing of the graph in the library

The pl command can be used in the author mode only.

(Note: DO NOT DELETE funfun)

GO TO ACTION 16,TABLE 1.1

16.0

Note the graphs in the library (including funfun). Now create a graph in your terminal buffer. Recall that this involves setting the x and y scale values, defining the axis origin, and defining coordinates for the desired points (refer to Table 1.1 if necessary). The name of your graph can be from 1 to 6 characters in length and include any of the characters (except the "enter" character) in the system dictionary. Call your graph "my own", and

GO TO ACTION 17, TABLE 1.1

17.0

Your own "my own" is in the graph library. Any author can now use

fn trnsfr °/ my own

to bring that graph into the student's terminal buffer.

GO TO ACTION 18, TABLE 1.1

18.0

Now the graph "my own" is no longer available in the library.

You may wish to try out various combinations of the features to which you have been introduced. One word of caution: PLEASE DO NOT DELETE ANY GRAPHS FROM THE LIBRARY OTHER THAN THOSE YOU CREATE.

It is hoped that extended features can be added to the demo program periodically. If you have any comments or suggestions for improvement related to the graph functions, the demo program, or this documentation, please make these on the attached blank sheets. Include your name if you would like to discuss these comments in more detail.

A P P E N D I X B

LISTING OF DEMONSTRATION PROGRAM "PLOT"

```
**PLOT  000                                        00202
PLOT*E
     1 PRR  *E
     2 DE   0+/32*E
     3 DT   0,0+/2,0+/40,0+/(T)HIS AREA*E
     4 DT   ,0+/2,+/40,0+/IS SAVED*E
     5 DT   ,0+/2,+/40,0+/FOR THE*E
     6 DT   ,0+/2,+/40,0+/AUTHOR*E
     7 DT   ,0+/2,+/40,0+/TO USE*E
     8 DT   ,0+/2,+/40,0+/FOR*E
     9 DT   ,0+/2,+/40,0+/EXPLAN-*E
    10 DT   ,0+/2,+/40,0+/ATORY*E
    11 DT   ,0+/2,+/40,0+/MATERIAL.*E
    12 DT   ,0+/2,+/40,0+/---------*E
    13 DT   ,0+/2,+/40,0+/---------*E
    14 DT   ,0+/2,+/40,0+/---------*E
    15 DT   ,0+/2,+/40,0+/---------*E
    16 DT   ,0+/2,+/40,0+/---------*E
    17 DT   ,0+/2,+/40,0+/---------*E
    18 DT   ,0+/2,+/40,0+/----------------------------------------*E
    19 FN   PLOT*E
EH*E
     1 DT   30,25+/2,30+/10,25+/(W)HAT NEXT(/*E
     2 EPI  30,36+/2,30+/1,36+/+/1+/*E
     3 CA   S+/CC*E
     4 BR   SETAX*E
     5 CA   A+/CC*E
     6 BR   AXIS*E
     7 CA   T+/CC*E
     8 BR   TRNSF*E
     9 CA   D+/CC*E
    10 BR   DELETE*E
    11 CA   P+/CC*E
    12 BR   PLOT*E
    13 CA   M+/CC*E
    14 BR   POINT*E
    15 AA   (8+/)AA*E
    16 BR   EH*E
SETAX*E
     1 PR   *E
     2 DE   0+/32*E
     3 DT   ,0+/2,+/40,0+/(G)IVE X AXIS START VALUE.*E
     4 EP   ,0+/2,+/40,0+/+/99+/*E
     5 AA   (8+/)AA*E
     6 LD   0+/C23*E
     7 FN   FCALC+/5*E
     8 BR   SETAX+/C23+/G+/0*E
     9 FN   SET+/XS*E
SETXE*E
     1 DT   ,0+/2,+/40,0+/(G)IVE X AXIS END VALUE*E
     2 LD   0+/C23*E
     3 EP   ,0+/2,+/40,0+/+/99+/*E
     4 AA   (8+/)AA*E
     5 FN   FCALC+/5*E
     6 BR   SETXE+/C23+/G+/0*E
```

```
      7 FN   SET+/XE*E
SETYS*E
      1 DT   ,0+/2,+/40,0+/(G)IVE (Y )AXIS START VALUE *E
      2 LD   0+/C23*E
      3 EP   ,0+/2,+/40,0+/+/99+/*E
      4 AA   (8+/)AA*E
      5 FN   FCALC+/5*E
      6 BR   SETYS+/C23+/G+/0*E
      7 FN   SET+/YS*E
SETYE*E
      1 DT   ,0+/2,+/40,0+/(G)IVE (Y )AXIS END VALUE*E
      2 LD   0+/C23*E
      3 EP   ,0+/2,+/40,0+/+/99+/*E
      4 AA   (8+/)AA*E
      5 FN   FCALC+/5*E
      6 BR   SETYE+/C23+/G+/0*E
      7 FN   SET+/YE*E
      8 BR   PLOT*E
POINT*E
      1 PR   *E
      2 DE   0+/32*E
      3 DT   ,0+/2,+/40,0+/(E)NTER (X )COORDINATE VALUE*E
      4 EP   ,0+/2,+/40,0+/+/99+/*E
      5 LD   0+/C22*E
      6 LD   0+/C23*E
      7 FN   FCALC+/5*E
      8 BR   POINT+/C23+/G+/0*E
      9 LD   C30+/C28*E
     10 LD   C29+/C27*E
POINT2*E
      1 PR   *E
      2 LD   0+/C22*E
      3 LD   0+/C23*E
      4 DT   ,0+/2,+/40,0+/(G)IVE (Y )COORDINATE VALUE*E
      5 EP   ,0+/2,+/40,0+/+/99+/*E
      6 FN   FCALC+/5*E
      7 BR   POINT2+/C23+/G+/0*E
      8 FN   ADDPNT*E
      9 BR   PLOT*E
AXIS*E
      1 PR   *E
      2 DE   0+/32*E
      3 DT   ,0+/2,+/40,0+/(D)O YOU WISH TO DELETE AXES(/*E
      4 EP   ,0+/2,+/40,0+/+/3+/*E
      5 CA   YES+/CC*E
      6 FN   AXIS+/D*E
      7 BR   PLOT*E
      8 AA   (8+/)AA*E
AXISX*E
      1 DT   ,0+/2,+/40,0+/(E)NTER (X )COORDINATE OF (Y )AXIS.*E
      2 LD   0+/C23*E
      3 EP   ,0+/2,+/40,0+/+/99+/*E
      4 AA   (8+/)AA*E
      5 FN   FCALC+/5*E
      6 BR   AXISX+/C23+/G+/0*E
```

```
      7 LD   C30+/C28*E
      8 LD   C29+/C27*E
AXISY*E
      1 DT   ,0+/2,+/40,0+/(E)NTER (Y )COORDINATE OF (X )AXIS. *E
      2 LD   0+/C23*E
      3 EP   ,0+/2,+/40,0+/+/99+/*E
      4 AA   (8+/)AA*E
      5 FN   FCALC+/5*E
      6 BR   AXISY+/C23+/G+/0*E
      7 FN   AXIS*E
      8 BR   PLOT*E
DELETE*E
      1 PRR  *E
      2 FN   DELETE*E
      3 DE   0+/32*E
      4 DT   30,0+/2,30+/40,0+/COMMAND COMPLETED*E
      5 PAE  *E
      6 BR   PLOT*E
TRNSF*E
      1 DE   0+/32*E
      2 DT   ,0+/2,+/40,0+/(E)NTER PLOT NAME.*E
      3 EP   ,0+/2,+/40,0+/+/6+/*E
      4 FN   MV+/'        '+/B0+/A*E
      5 WA         (8+/)WW*E
      6 BR   TRNSF*E
      7 CA   (8+/)CC*E
      8 FN   TRNSFR*E
      9 BR   PLOT*E
     10 EN   *E
**410-11 READER NOT READY
**410-12 LAST CARD READ
**410-20 END OF JOB
```